

# Automatización y gestión del H2O. Control Remoto



## 1- Usos/casos/funciones

Introducción a la materia a partir de situaciones del entorno eco-industrial.

Paseo a el caso práctico calafou, casuística calafou.

## 2- Dispositivos electrónicos

Hardware : sensores, actuadores y microprocesadores.

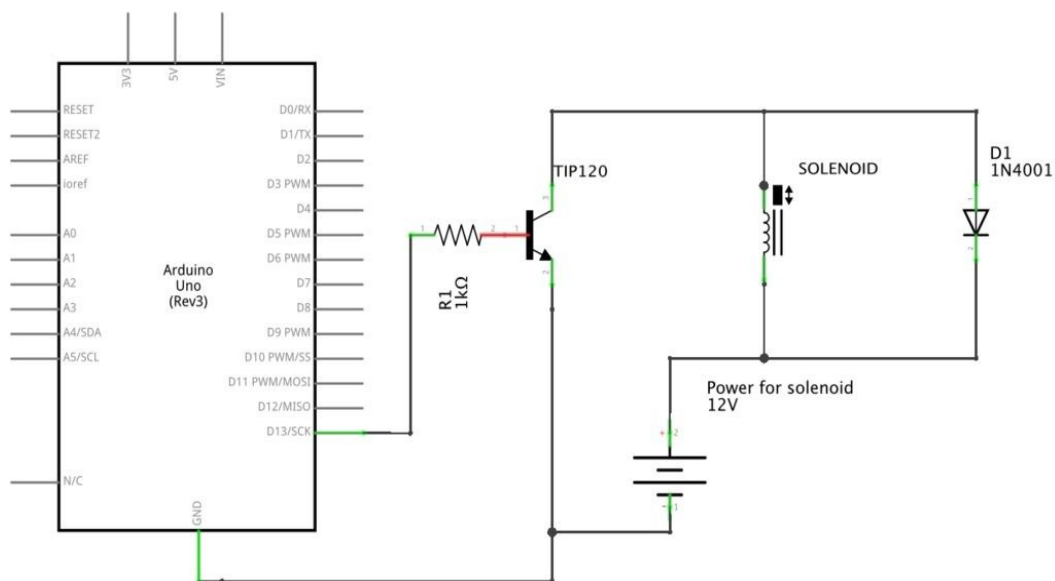
# Sensores

Controlar el flujo de fluidos utilizando el flujo de electrones!

## Electroválvula

Normalmente, la válvula está cerrada. Cuando 12 VDC se aplica a los dos terminales, la válvula se abre y el agua pasa a través. La válvula tiene una disposición de junta en el interior, por lo que no es un requisito presión mínima de 0,02 Mpa (3 PSI). Además, el líquido puede fluir sólo en una dirección. Probamos el solenoide con diversos voltajes DC y encontramos que pudimos accionar hacia abajo en 6VDC (aunque era un poco más lento para abrir). Sugerimos un TIP120 o FET de potencia N-Channel con un diodo 1N4001 para accionar esto desde un pin del microcontrolador.

Una alimentación 9V 1A o 12V 1A es suficiente para nuestro setup

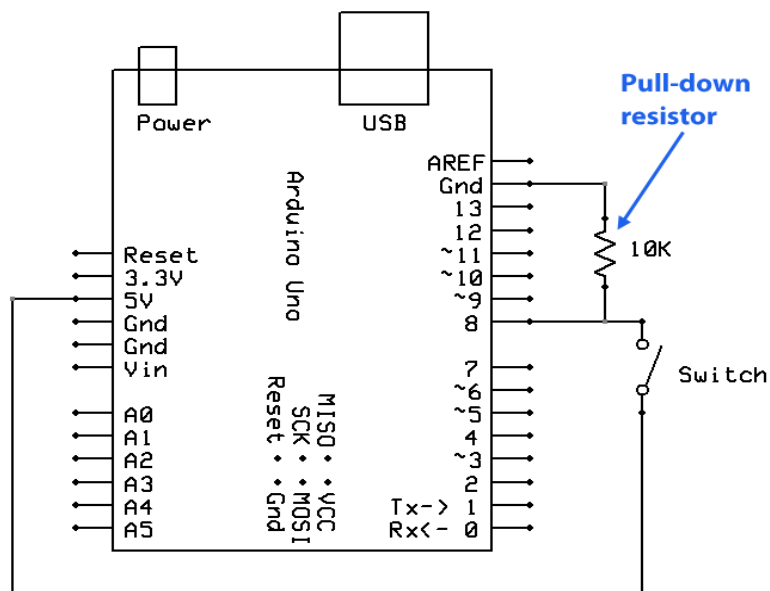


fritzing

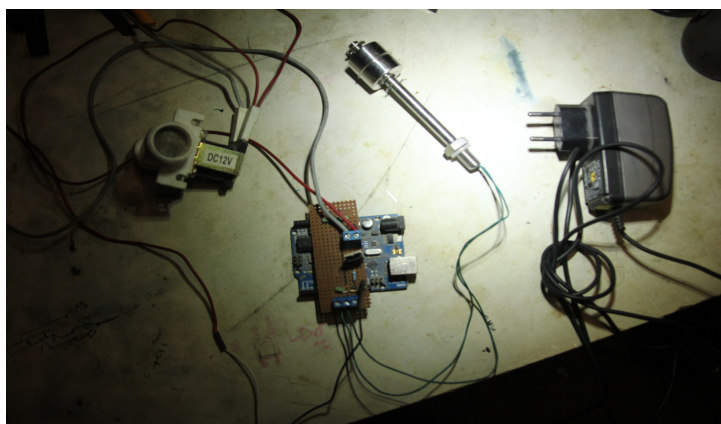
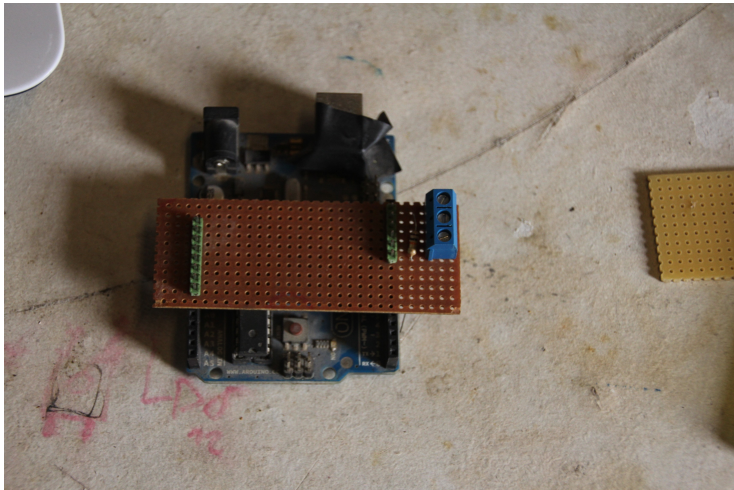
\_ (sensor switch, bolla)



El siguiente circuito agrega un "pull-down" resistencia de 10K (el valor exacto no importa, siempre y cuando no sea demasiado baja).



Esta resistencia "débilmente" tira del interruptor hacia abajo a la tierra, de modo que si el interruptor está abierto, tendrá 0V en él (a través de la resistencia) y así se registrará bajo si no se presiona, y HIGH si se presiona.



Usando un TRANSISTOR en lugar de Relés( normalmente para bajos voltajes de corriente continua)

El **transistor de efecto de campo metal-óxido-semiconductor** o **MOSFET** (en inglés *Metal-oxide-semiconductor Field-effect transistor*) es un [transistor](#) utilizado para amplificar o conmutar [señales](#) electrónicas. Es el transistor más utilizado en la industria microelectrónica, ya sea en circuitos analógicos o digitales, aunque el [transistor de unión bipolar](#) fue mucho más popular en otro tiempo. Prácticamente la totalidad de los [microprocesadores](#) comerciales están basados en transistores MOSFET.

El MOSFET es un dispositivo de cuatro terminales llamados surtidor (S), drenador (D), compuerta (G) y sustrato (B). Sin embargo, el sustrato generalmente está conectado internamente al terminal del surtidor, y por este motivo se pueden encontrar dispositivos MOSFET de tres terminales.

El término 'metal' en el nombre MOSFET es actualmente incorrecto ya que el material de la compuerta, que antes era metálico, ahora se construye con una capa de [silicio policristalino](#). El aluminio fue el material por excelencia de la compuerta hasta mediados de 1970, cuando el [silicio policristalino](#) comenzó a dominar el mercado gracias a su capacidad de formar compuertas auto-alineadas. Las compuertas metálicas están volviendo a ganar popularidad,

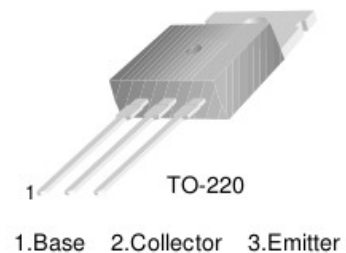


dada la dificultad de incrementar la velocidad de operación de los transistores sin utilizar componentes metálicos en la compuerta. De manera similar, el 'óxido' utilizado como aislante en la compuerta también se ha reemplazado por otros materiales con el propósito de obtener canales fuertes con la aplicación de tensiones más pequeñas.

## TIP120/121/122

### Medium Power Linear Switching Applications

- Complementary to TIP125/126/127



### NPN Epitaxial Darlington Transistor

**Arduino** es una plataforma de [hardware libre](#), basada en una [placa](#) con un [microcontrolador](#) y un [entorno de desarrollo](#), diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.[2](#) [3](#)

El [hardware](#) consiste en una placa con un microcontrolador [Atmel AVR](#) y puertos de [entrada/salida](#).[4](#) Los microcontroladores más usados son el [Atmega168](#), [Atmega328](#), [Atmega1280](#), y [Atmega8](#) por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños. Por otro lado el software consiste en un entorno de desarrollo que implementa el [lenguaje de programación Processing/Wiring](#) y el [cargador de arranque](#) que es ejecutado en la placa.

Desde octubre de 2012, Arduino se usa también con microcontroladoras CortexM3 de ARM de 32 bits,[5](#) que coexistirán con las más limitadas, pero también económicas AVR de 8 bits. ARM y AVR no son plataformas compatibles a nivel binario, pero se pueden programar con el mismo IDE de Arduino y hacerse programas que compilen sin cambios en las dos plataformas. Eso sí, las microcontroladoras CortexM3 usan 3,3V, a diferencia de la mayoría de las placas con AVR que generalmente usan 5V. Sin embargo, ya anteriormente se lanzaron placas Arduino con Atmel AVR a 3,3V como la Arduino Fio y existen compatibles de Arduino Nano y Pro como Meduino en que se puede conmutar el voltaje.

Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software tal como [Adobe Flash](#), [Processing](#), [Max/MSP](#), [Pure Data](#). Las placas se pueden montar a mano o adquirirse. El [entorno de desarrollo integrado](#) libre se puede descargar gratuitamente.

Arduino puede tomar información del entorno a través de sus entradas analógicas y digitales, puede controlar luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un computador.

El proyecto Arduino recibió una mención honorífica en la categoría de Comunidades Digital en el [Prix Ars Electrónica](#) de 2006.

La demo de interacción entre electroválvula y arduino la hemos realizado utilizando el ejemplo blink que viene por defecto en su framework de programación. Está pensado para LED, xro funciona en cualquier caso de switch.

<http://www.arduino.cc/en/Tutorial/Blink?action=sourceblock&num=1>

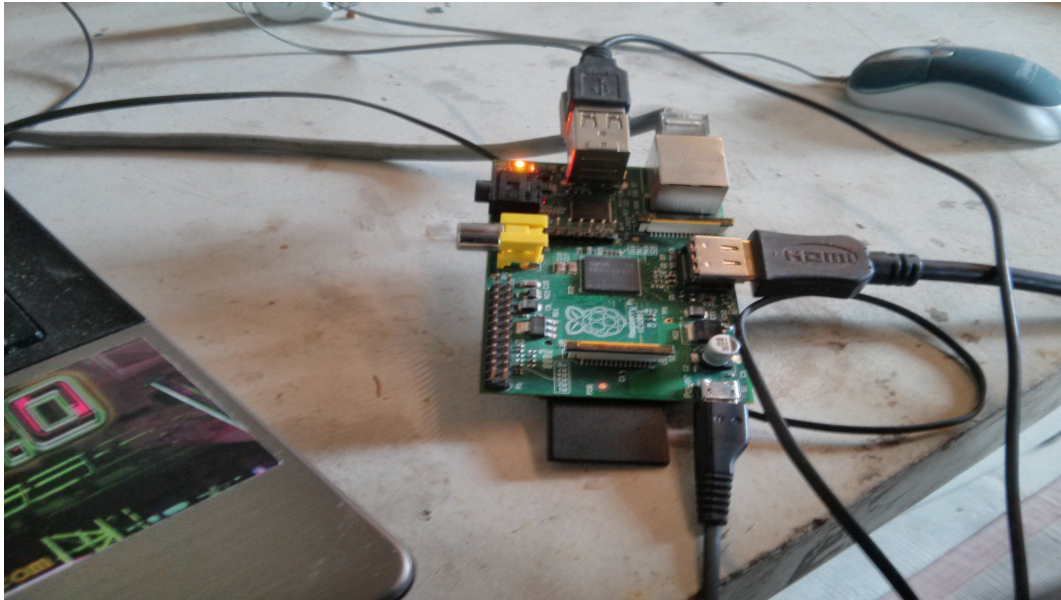
Links de referencia:

<http://www.adafruit.com/products/997>

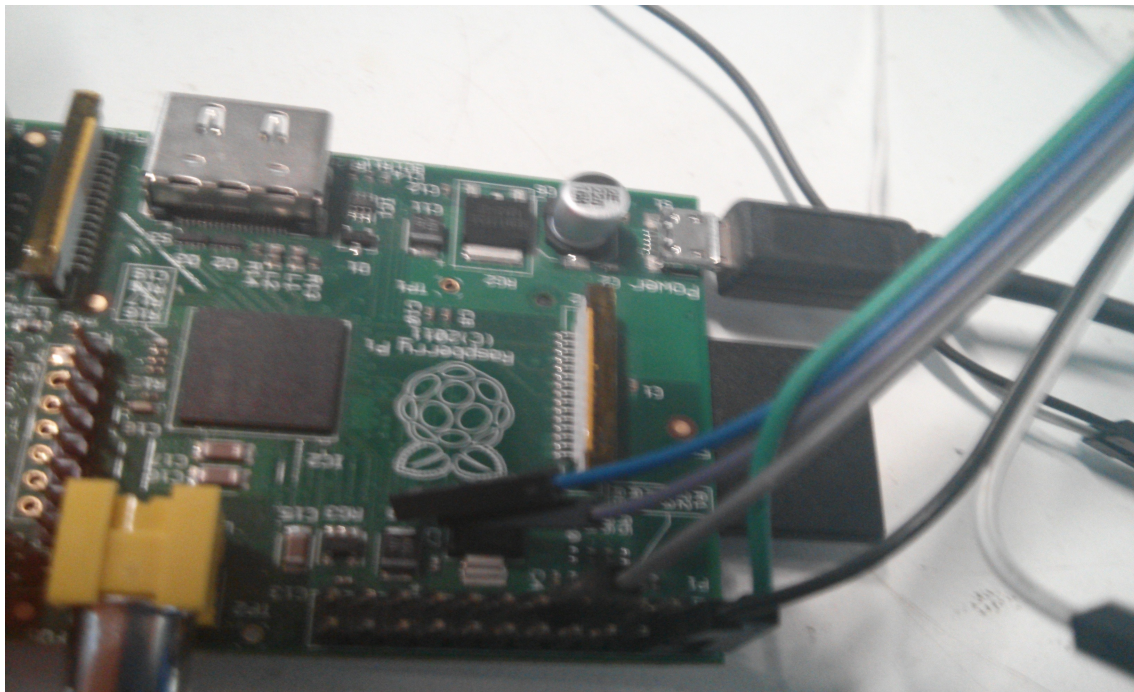
<http://electronics.stackexchange.com/questions/122908/controlling-solenoid-valve-using-raspberry-pi>

# Dispositivos autónomos (utilizando la Raspberry)

## Raspberry Modelo B



Para realizar la demo de control remoto del relay (switch) de la electroválvula, nos hemos decantado por la Raspberry Pi ya que incorpora los GPIO pines que interactúan con el mundo físico como los pines de arduino, y además permite gestionar un webserver para habilitar el control remoto sin necesidad de añadir ningún tipo de shield o módulo. Su mayor capacidad en RAM etc, lo convierte en una opción óptima para futuras implementaciones y desarrollos.





**Raspberry Pi es un ordenador de placa reducida o (placa única) (*SBC*) de bajo coste, desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.**

El diseño incluye un System-on-a-chip Broadcom BCM2835, que contiene un procesador central (CPU) ARM1176JZF-S a 700MHz (el firmware incluye unos modos "Turbo" para que el usuario pueda hacerle overclock de hasta 1 GHz sin perder la garantía), un procesador gráfico (GPU) VideoCore IV, y 512 MB de memoria RAM (aunque originalmente al ser lanzado eran 256 MB). El diseño no incluye un disco duro ni unidad de estado sólido, ya que usa una tarjeta SD para el almacenamiento permanente; tampoco incluye fuentes de alimentación ni carcasa. El 29 de febrero de 2012 la fundación empezó a aceptar órdenes de compra del modelo B, y el 4 de febrero de 2013 del modelo A.

La fundación da soporte para las descargas de las distribuciones para arquitectura ARM, Raspbian (derivada de Debian), RISC OS 5, Arch Linux ARM (derivado de Arch Linux) y Pidora (derivado de Fedora); y promueve principalmente el aprendizaje del lenguaje de programación Python.5 Otros lenguajes también soportados son Tiny BASIC, 12 C, Perl5 y Ruby (y Javascript, el lenguaje de programación por excelencia en la web).

**-Forzar el conector HDMI para que se inicie automáticamente y poder instalar.**

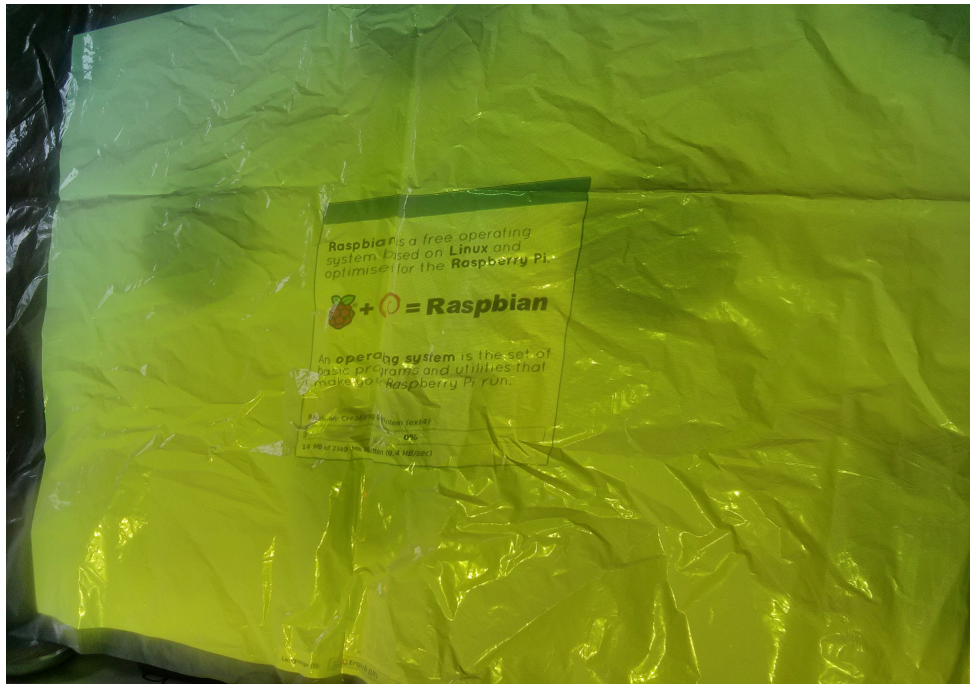
[http://elinux.org/RPiconfig#cite\\_note-hdmi\\_edid\\_file-1](http://elinux.org/RPiconfig#cite_note-hdmi_edid_file-1)

<http://blog.goshield.es/2013/03/problema-mi-raspberry-pi-no-saca-video.html>

**-Instalar Rapsbian.**



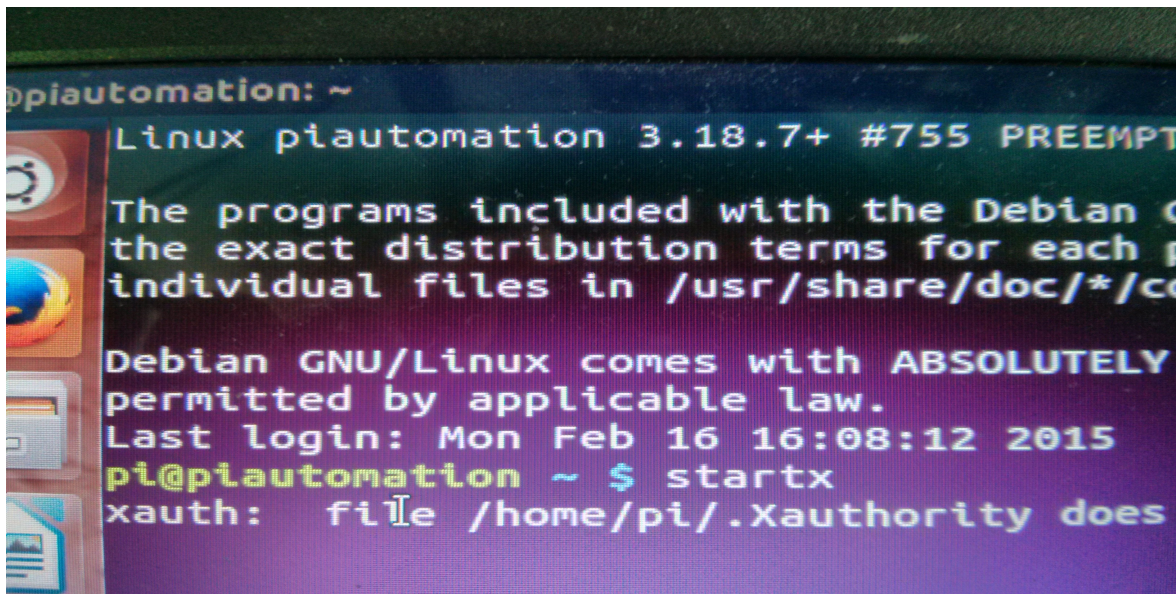




<http://www.ubuntufacil.com/2014/02/primeros-pasos-con-raspberry-pi-instalacion-y-configuracion-de-raspbian/>

**-Al instalar se abre el raspi-config para configurar sistema**

**-Conectarse por ssh/podéis usar Filezilla**



Una vez habilitado con el raspi-config la conexión ssh, sólo tendremos que averiguar la dirección IP de la Raspberry con un ifconfig -a para poder conectarnos desde una terminal o desde un programa para conectarse vía FTP como Filezilla.

<https://filezilla-project.org/>

### **-Instalar Node.js.**

Node.js es una plataforma construida con Javascript runtime que permite construir de manera simple y rápida aplicaciones escalables en red. Su tecnología basada en eventos con un modelo I/O sin bloqueos le permite ser muy poco “pesada” y muy eficiente, permitiendo extender sus funcionalidades en base a módulos de sencilla instalación. Por todo ello, se está convirtiendo en casi un standard para el Open Home Automation, la Open Ecology y el Internet of Things cuando existe control remoto. Al estar basada en javascript, su customización se realiza de manera sencilla con html y css, permitiendo incorporar diseño responsivo adaptado a los diferentes dispositivos móviles.

Hemos instalado la versión node.js v.0.10, dado que la versión existente en el github daba un error frecuente en raspberry (por lo que hemos visto en foros) de no reconocer el npm install (comando necesario para instalar módulos) posteriormente.

<http://blog.nodejs.org/2015/01/26/node-v0-10-36-stable/>

Una vez descargada, añadimos al PATH las rutas de node en un fichero, por ejemplo .bash\_profile.

```
export NODE_JS_HOME=/home/pi/node-v0.10.36-linux-arm-pi
export PATH=$PATH:$NODE_JS_HOME/bin
```

## **-Instalar el gpio-admin package.**

Desde terminal:

```
git clone git://github.com/quick2wire/quick2wire-gpio-admin.git
cd quick2wire-gpio-admin
make
sudo make install
sudo adduser pi gpio
```

Es necesario desloguearse y volver a loguearse para activarlo.

## **-Instalar el pi-node-relay del Open Home Automation.**

Desde terminal:

```
git clone https://github.com/openhomeautomation/pi-node-relay
```

## **-Instalar los módulos necesarios de node.js**

Desde terminal y una vez dentro de la carpeta pi-node-relay, ejecutar el comando:

```
sudo npm install pi-gpio express
```

El módulo express servirá para crear el webserver. El módulo pi-gpio para que el node se comunique directamente con los pines de la Raspberry.

Cabe decir que existen otros módulos como cylon y cylon-gpio cylon-raspi y cylon-i2c que también hemos experimentado conjuntamente a express, decantándonos de entrada por pi-gpio para la automatización con control remoto.

En la estructura de pi-node-relay hay una carpeta public dónde se encuentra el html de la interface y una carpeta views dónde se encuentran los css y js de la customización de diseño. El core de lo que sucede está en el app.js dónde se llama a los diferentes módulos, se crea el webserver y se define el puerto de salida (el 3700 en este caso), se crean rutas de acceso del server a la interface y al mapping de los gpio-pines, se imprimen la respuesta a las peticiones de acceso para facilitar el debugging y se envía la orden de abrir y cerrar el relay. Se cierra conexión con browser y se inicializa server.

En la interface html se definen los botones y la interactividad de activar y desactivar el relay con el evento onclick.

## **-Inicializar la app.**

Desde la carpeta pi-node-relay, ejecutar el comando node app.js para inicializar la app.

Documentación de referencia:

<https://www.openhomeautomation.net/control-a-relay-from-anywhere-using-the-raspberry-pi/>

<https://carlosazaustre.es/blog/programando-en-node-js-con-raspberrypi-primeros-pasos/>

Ahora tenemos accesible un control remoto que nos permite activar y desactivar el relay. De esa manera, podemos dejar la electroválvula en estado abierto o en estado cerrado el tiempo que deseamos, modificándolo con un simple click desde cualquier lugar de nuestra red local, sin desplazarnos hasta el lugar físico exacto. Este era el objetivo inicial del cual surgió el nodo pensando en la casuística de Calafou.

### **-Acceder al control remoto desde internet.**

Para hacer accesible el control remoto desde cualquier punto más allá de la red local, es decir desde internet directamente, podemos hacer diferentes cosas. Recomendamos estas dos:

- 1.** En modo desarrollo y test de prototipo: ejecutar dentro de pi-node-relay el comando `npm install -g localtunnel` y, una vez instalado este módulo, el comando `lt --port 3700`. Esto generará un túnel hacia el server y te devolverá una URL a la que te podrás conectar desde internet para activar y desactivar el relay. El inconveniente para utilizar este método en modo producción es la seguridad.
- 2.** En modo producción: abrir el puerto correspondiente en tu router, asociándolo a la IP de tu local server. No es en absoluto perfecto a nivel seguridad, deja una puerta abierta igualmente, pero es mejor opción.

**El desarrollo de diferentes implementaciones del prototipo continuará...;)**

**pineada+geekshabeka**



